Xiaolong Zhang
School of Information Resources and Library Science
University of Arizona
Tucson AZ 85719 USA

Correspondence to
xiaolong@u.arizona.edu

George W. Furnas
School of Information
University of Michigan

mCVEs: Using Cross-Scale Collaboration to Support User Interaction with Multiscale Structures

Abstract

In this paper, a new type of interaction environment, the multiscale collaborative virtual environment (mCVE), is proposed to support multiple users working together at different scale levels. This paper introduces the concept of multiscale collaboration in the context of 3D virtual environments and describes the benefits of multiscale collaboration for understanding and managing large structures that present important features at different scale levels. After a discussion on the design and implementation of multiscale tools to support the visualization of structures, cross-scale information sharing, and cross-scale action, the paper presents an experimental study showing that cross-scale collaboration can improve user performance.

1 Introduction

We live in a world where objects can span many different scale levels. Matter in the real world demonstrates various structures and characteristics at different length-scale levels. For example, images of the Earth at different scales range from a billiard ball in empty space (Figure 1a), to a sphere covered by continents and seas (Figure 1b), to a flat plane with different terrains (Figure 1c). DNA, observed at different scales, could appear as coiled DNA strands (Figure 2a), as the double helix of a DNA strand (Figure 2b), or as molecular building blocks (Figure 2c). These structures all exhibit important characteristics at different scale levels, and can be called *multiscale structures.*

Our interests in the world span numerous scales, and cross-scale approaches are often seen. In materials science, for example, scientists need cross-scale structural models, from nanometer to millimeter, to understand the characteristics of materials from the atomic level (e.g., the strength of atomic bonds) to the macroscopic level (e.g., mechanical stress) (Robbins, 2001). In architectural design and urban planning, new things need to be built with the consideration of larger contexts that embed these things and smaller structures below them (Alexander, Ishikawa, & Silverstein, 1977). Creating a new commercial district may require people to examine how the new district can fit the whole city at the city level, how individual buildings should be designed to match the style of the whole street at the street level, and how changes at one level may affect the design at others (e.g., how opening a new route at the street level

Presence, *Vol. 14, No. 1, February 2005,* 31–46

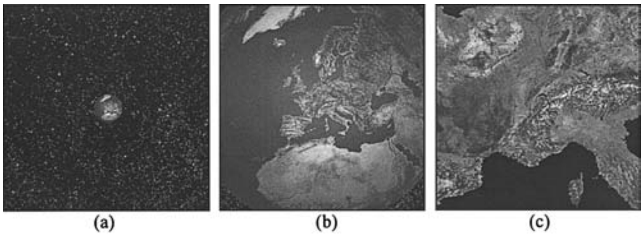


Figure 1. *Multiscale structures of the Earth (Courtesy of CERN).*

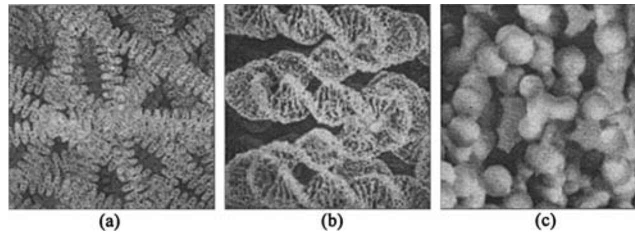


Figure 2. *Multiscale structures of DNA (Morrison & Morrison, 1994).*

may change the design coherence at the district level). To complete such multiscale tasks, we need tools to control the scale level of observation and analysis so that we can see structures at different levels and understand their relationships. This is a challenge, given that the normal interaction scale range of human beings is only about five orders of magnitude, from millimeters (10^{-3} m) to hundreds of meters (10^2 m). The size scale range of matter in the real world is much larger, spanning about forty-two orders of magnitude, from 10^{-16} m to 10^{26} m (Morrison & Morrison, 1994).

We use many different tools in real life to help us cross scale boundaries. For example, we have microscope pictures, satellite pictures, and scaled model structures for the observation and analysis of objects or structures beyond our normal interaction scale range (e.g., DNA structures, architectural structures, planet systems). However, these artifacts give us only static information at a particular scale level. It is still difficult to receive information from other scale levels. Even when pictures or model structures at different scales are available, interpreting how these pictures or models are related is still a challenge. Computer tools for modeling suffer from similar problems because of the lack of support for multiscale interaction. For example, Cerius^2, a very powerful tool to simulate small and ideal structures (e.g., unit cells of organic thin films) in materials science, can hardly go beyond the molecular level for larger structures (e.g., a structure with $100 \times 100 \times 100$ unit cells) that are closer to real-world situations. In architectural design, complex large-scale projects, such as airports, have many thousands of interconnected designs at different scales. However, existing computer-aided tools are very limited in support of the visualization, organization, and navigation of these designs (A. Summerfield, personal communication, December 12, 2003).

Such a challenge in handling multiscale structures also exists when people work with information structures in virtual worlds. Information structures such as file systems, digital libraries, and the World Wide Web are rapidly becoming larger and larger. These structures can easily have more than tens of thousands of components, overwhelming our memories and information-processing capabilities. We usually deal with large structures in a hierarchical way, either explicitly or implicitly. For example, file systems are organized as explicit hierarchies. The semantic structure that bonds the maps and related geographic information in Figure 3 uses hierarchy implicitly.

Hierarchies help us focus on issues within an appropriate scale level, but in limiting our attention to a certain scale range we will lose some important information of large structures. Either detailed content or global context information has to be hidden. For example, in Windows, we often use an Explorer window to display a file system with a balanced view of the file structure in the left pane and the contents of a directory in the right. However, for large file systems, this window cannot tell users where the directory of interest is actually located in the hierarchy. Although the scroll bars are helpful for browsing the directory list, deciphering a large structure from a long list can be difficult for many users. In a geographic-information system such as Mapquest, users can jump to different maps for scale-related informa-

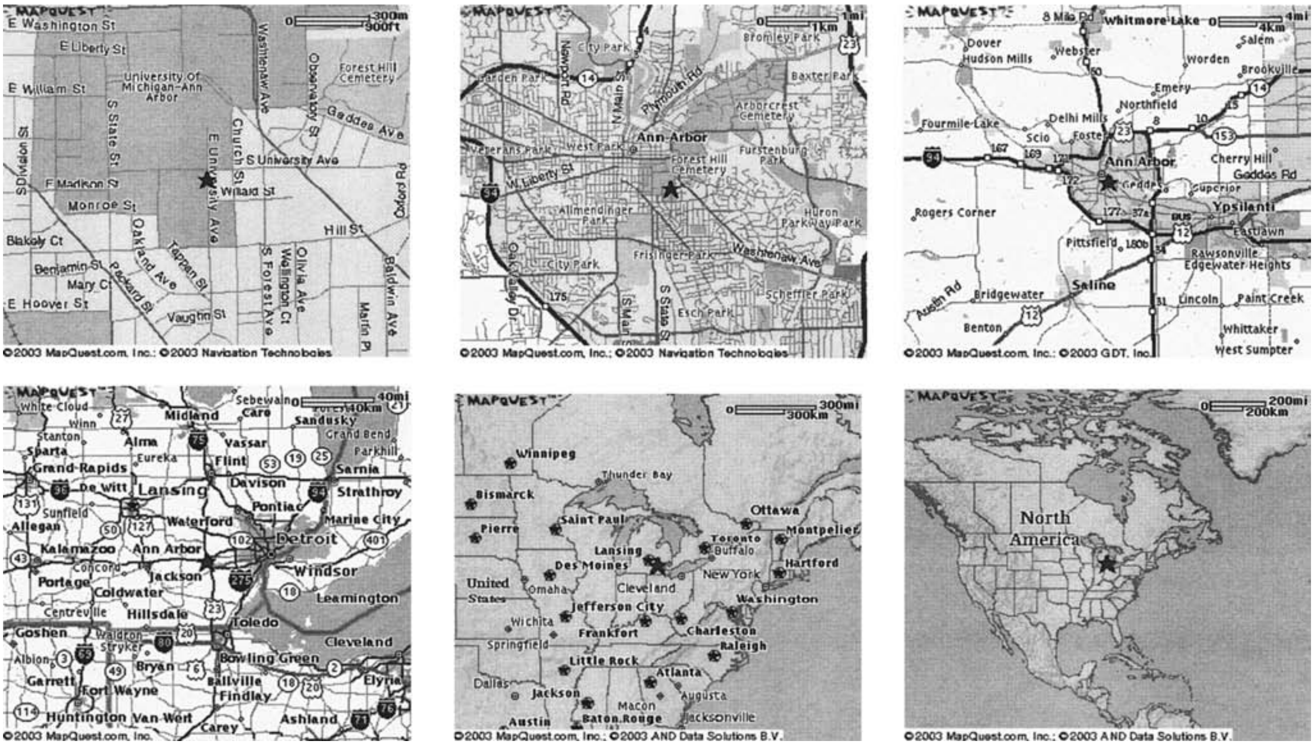


Figure 3. *Maps of a US city at different scales (Images provided by www.mapquest.com).*

tion, as seen in Figure 3, but making sense of the relationship between these different maps across scales may demand considerable cognitive resources for memory and information processing. Users need help controlling the scale levels of observation and analysis.

To address these multiscale issues, we propose a new type of interaction environment, the multiscale collaborative virtual environment (mCVE). Multiscale technology allows people to control their observation and action domains dynamically. Collaboration technology benefits people because of the division of labor and the parallel working processes. Combining multiscale and collaboration helps to better allocate individual users' cognitive resources to different scales and then exploit the resulting different perception and action capabilities. This multiscale collaboration technique is discussed in the context of 3D virtual environments, which can visualize more complicated structures. mCVEs allow users to work at different scale levels, being the size of ants to observe fine details and maintain high action accuracy and being the size of giants to see the big picture of structures and have a broad interaction range. By collaborating from different scale levels in mCVEs, users can bridge the scale barrier.

This paper begins with the introduction of mCVEs and literature review. Then it advances to design issues related to the visualization of multiscale structures, cross-scale information sharing, and cross-scale action. After a discussion of the implementation of an mCVE prototype system, the paper describes an experimental study of the effectiveness of multiscale collaboration.

2 Multiscale Collaborative Virtual Environments

The design of mCVEs deviates from the path set by Ivan Sutherland, which is to make computer-simulated worlds real (Sutherland, 1965). By going beyond being real (Hollan & Stornetta, 1992; Stappers,

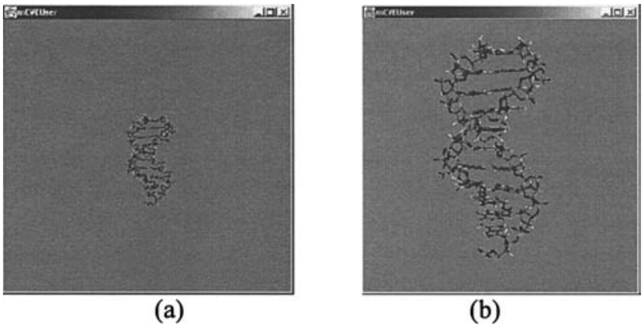


Figure 4. *DNA rendered at two different scales.*

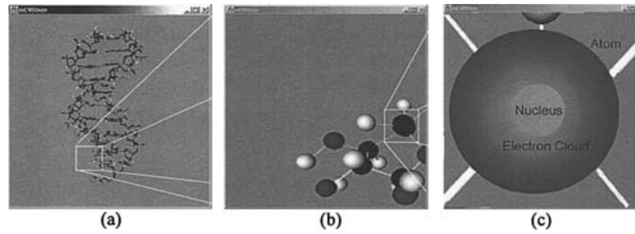


Figure 5. *Successive views of a DNA structure rendered at different scales. Perspective lines added for the figure to show where each successive view comes from.*

Gaver, & Overbeeke, 2000), we can expand design space and find new ways to address old problems. This section discusses the features of multiscale virtual environments (mVEs), the necessity of multiscale collaboration, and the benefits of mCVEs. Relevant literature is also reviewed.

2.1 Multiscale Virtual Environments

The absence of real-world physics in virtual environments has led to some innovative designs in virtual environments. Teleportation, for example, is a useful navigation tool that transcends speed and time. An mVE is a design that overcomes the scale barrier and allows users to manipulate the scale of the virtual space in their work.

2.1.1 Scale in Virtual Environments. In computer graphics, scale usually refers to the rendered size of objects. To manipulate the scale of a virtual environment is to resize the virtual environment. Figure 4 shows images of a DNA structure at two scales. These two images differ only in rendered sizes and resulting different levels of detail. However, if the scale difference is more dramatic, image difference can go beyond geometric size, as seen in Figure 5, where three images at significantly different scales show very different and important characteristics: the double helix in (a), the molecular structure in (b), and the nucleus and the electron cloud in an atom in (c).

2.1.2 Scale in Multiscale Virtual Environments. This scale-dependent rendering technique distinguishes mVEs from conventional virtual environments. Conventional virtual environments render objects with the same representation, regardless of user interests. mVEs render objects with different representations corresponding to different scale ranges. In mVEs, users can easily change their observation scales and receive different representations, from big pictures to local details.

Another important but less obvious characteristic of mVEs is that users also get different interaction domains. To change the scale of a virtual environment is to shift a user's action capabilities. One action can lead to different results at different scales. For example, a user's movement could be at the nanometer level when the user is at the atomic scale, and the same movement would be at the micrometer level when the user moves to the molecular scale. When the user observes a virtual planet system, the movement scale can be at the level of hundreds of thousands of kilometers. In mVEs, user observation and action capabilities are coordinated at the same or comparable scale level.

Users need this association in mVEs so that they can work appropriately with what they see. For example, when users observe an object structure at the level of 1 mm, their movement scale should be at a comparable scale, say 1 mm/sec. Moving at a very different scale, say 1 km/sec, would make it hard to keep a consistent view. Objects of interest would be constantly speeding out of sight. Conversely, if the structure of interest is at the level of 1 km, moving at a scale of 1 mm/sec would be problematic. It would demand tremendous effort to change the view of the large structure even slightly. Other interaction parameters such as reachable distance and manipulation accuracy should be similarly coordinated.

Thus, mVEs provide users with new interaction capabilities that are usually unavailable in conventional virtual environments. In mVEs, users can interact with virtual worlds at multiple scale levels. At any given scale, some things will be easy to see and manipulate, while others will be either too large or too small to display. Changing the scale moves new things into the easily visible and manipulative range. Coordinated observation and action capabilities help users to better interact with multiscale structures. Multiscale extends the design space for information visualization and user interaction.

2.2 Multiscale Collaboration

Multiscale collaboration can further facilitate user interaction with multiscale structures. In particular, it can help people with work on large structures that need more labor, on rapidly changing structures that demand parallel working, and on complex structures that require domain knowledge beyond what an individual may have.

2.2.1 Large Multiscale Structures. Managing a large multiscale structure could be difficult for a single user. Take a city-planning example. Adding new design elements (e.g., paths or landmarks) may force planners to move back and forth frequently between different scales and check the impact of new additions on design at different levels. For a large city, continuously changing scale could be very demanding and costly if there is just one planner. Multiscale collaboration can offer some help by placing individual planners with different scales and having them focus on issues at their own scale level. Planners can then work independently on problems within their own scale scopes, and collaboratively deal with cross-scale issues. Collaboration divides big cross-scale problems into smaller within-scale ones that can be more easily handled by individuals.

2.2.2 Rapidly Changing Multiscale Structures. It could also be a challenge for a user to manage a structure that changes rapidly, however small that structure may be. Consider a scenario in which a small computer network, managed by a system administrator, is being infected by a worm virus. The system administrator may not be able to keep pace with the speed at which the virus spreads. If more system administrators are involved, they can have individuals to check subnets at different levels simultaneously, and then take appropriate actions to stop the virus from further damaging the entire network. Collaboration, by supporting parallel working, improves people's response to problems at different levels.

2.2.3. Cross-Domain Multiscale Structures. Challenges may also come from the diversified knowledge and expertise required in understanding and managing structures spanning across the boundaries of scientific domains. Understanding cross-domain multiscale structures could be daunting for individuals. For example, in studying the strength of a new material across different scale levels, no single person would possess all the required in-depth knowledge of chemistry, materials science, and mechanical engineering. Chemists, materials scientists, and mechanical engineers may need to work together on this issue. While they can focus on problems in their own expertise domains individually, working together helps them see how properties at different scale levels may affect each other and even discover important issues that might be ignored without cross-scale collaboration. Collaboration helps solve cross-scale problems by combining people's knowledge domains from different scales.

2.3 Multiscale Collaborative Virtual Environments

An mCVE is an mVE that supports collaboration. In mCVEs, users can choose their own scales in the observation of an object, and work together on the same object at different scales. A team of city planners, for example, can have a regular planner of the normal human size at street level, and a "giant" planner at the city

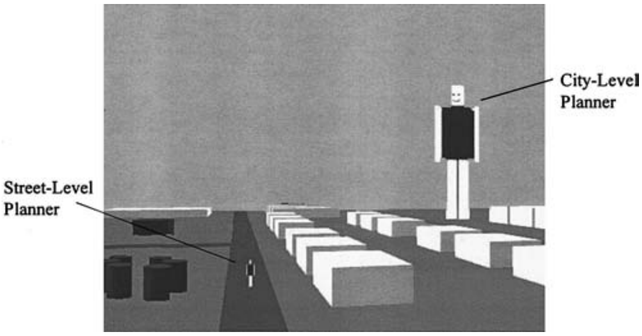


Figure 6. *Two city planners working at two different scales.*

level. While each planner focuses on planning issues at his or her own level, they can help each other on cross-scale issues. The giant planner can take the street-level planner quickly to a distant destination, and the street-level planner can help the partner place buildings or landmarks at their exact locations. Figure 6 shows two planners at two scales working together. Their cross-scale collaboration could help make a plan that meets the requirements at various levels.

As seen in Figure 6, each user is represented as an avatar in mCVEs, with a scaled size corresponding to the scale level of the user. This avatar metaphor offers two benefits. First, avatars provide rich awareness information (e.g., user view locations, viewing directions, body gestures) that are critical to social interactions in virtual environments (Benford, Bowers, Fahlén, Greenhalgh, & Snowdon, 1995). Second, by yoking different observation and action parameters under the same avatar body, avatars can help users better understand the association of different observation and action capabilities in mCVEs. The eye level of a user's avatar tells how far the user can see; the avatar's arm length indicates how far the user can reach; the leg length shows how fast the user can move. Seeing the size of a user's avatar, other users can tell at what scale the user is working and what the user can do.

mCVEs could be a promising tool to help people deal with multiscale structures, but this new design paradigm raises many research issues, ranging from the conceptual understanding of the implications of multiscale for user perception and action, to the technical design of mCVEs, to the experimental evaluation of the effectiveness of mCVEs.

2.4 Literature Review

Some research effort has been made to address design issues related to user interaction with large structures. Furnas (1986) proposed a theoretical framework, Generalized Fisheye Views, to manage large hierarchical structures by allocating more space for objects of interest. This technique has been used to visualize various types of data (Sarkar & Brown, 1992; Robertson & Mackinlay, 1993; Lamping, Rao & Pirolli, 1995; Raab & Ruger, 1996). Koike and Yoshihara (1993) used the fractal concept to expand the tree structure based on the degrees of interest. Cone Trees were built to present file hierarchies (Robertson, Mackinlay, & Card, 1991). These methods focused only on better using screen space to deal with the size of information structures, but gave little consideration to visualizing structures with different representations based on user interaction scales. Multiscale technology (Perlin & Fox, 1993; Bederson & Hollan, 1994; Lieberman, 1994) provided early inspiration for this research. This technique was usually regarded as an alternative user interface in 2D. This research extends it into 3D worlds and argues that multiscale technology is a powerful design that, by allowing people to control their observation and action domains at different scales, can augment our limited cognitive capabilities.

Scaling has been seen in some designs to support user actions in virtual environments. Scaling tools were used to increase reaching distance (Poupyrev, Billinghurst, Weghorst, & Ichikawa, 1996; Mine, Brooks, & Sequin, 1997; Pierce et al., 1997), facilitate spatial knowledge acquisition (Stoakley, Conway, & Pausch, 1995), and improve navigation performances (Mackinlay, Card, & Robertson, 1990). These scaling tools usually focused on only one interaction parameter, and ignored the implications of the change of one parameter for all other related perception and action. While isolating individual interaction parameters may work well in some situations, uncoordinated perception and action, as discussed previously, could cause some problems in the interaction with large multiscale structures.

Research literature on virtual environments supporting collaboration is massive, but consideration of cross-scale collaboration is rarely seen. The most relevant project is the CALVIN system (Leigh, Johnson, & DeFanti, 1996), in which two users can collaborate on a design project from two different perspectives. In these two perspectives, objects were rendered with same representations, and the system did not provide users with multiple levels of abstraction at different scales. Also, the system gave users limited choices of working scales, and could not satisfy their needs for active and interactive control over scale across a much broader range.

3 Design of mCVEs

There are some design guidelines for 3D collaborative virtual environments (CVEs) (Benford, Snowdon, Colebourne, O'Brien, & Rodden, 1997; EPFL et al., 1997; Tromp, 1999). However, they were constructed for the design of conventional CVEs and common collaboration actions in CVEs. Applying them in designing mCVEs could be difficult, because scale issues have not been considered in these guidelines. In this research, we adopt a design approach that starts from understanding the implications of multiscale technology for user interactions and then derives design considerations based on this understanding. In this section, efforts are made first to analyze how multiscale may affect user interaction. Then some conceptual designs are proposed to support user perception and action in mCVEs.

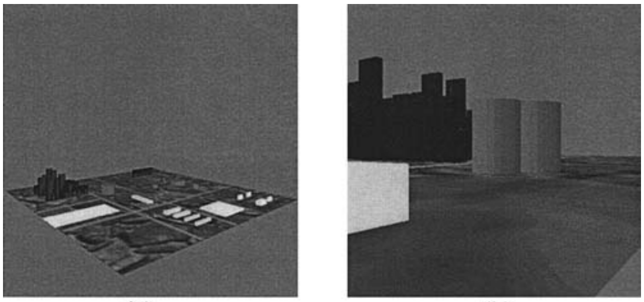
3.1 Implications of Multiscale for User Interaction

Multiscale could affect user interactions at different levels and in different ways. Here, we first discuss how multiscale can support the understanding of multiscale structures. Then, we describe how it may impede information sharing among users at different scales and how it may facilitate cross-scale action.

3.1.1 Understanding Multiscale Structures. As discussed previously, interacting with large structures requires users to obtain both detailed content and sufficient context information. This well-known "content + context" problem is indeed a multiscale problem, because detailed content information and global context information are usually distributed at different scale levels. To understand complicated structures that present different features at different scales, we usually break phenomena into components and focus our attentions on issues within certain levels rather than all levels at once (Pattee, 1973; Ahl & Allen, 1996), and then choose appropriate tools (e.g., microscopes, telescopes) for observation and analysis at different scales.

In virtual environments, multiscale technology could help address this "focus + context" problem by allowing users to control what they want to see and what they can see from multiscale structures, and then to manipulate the amount of context and content information displayed on the screen. Under different interaction scales, objects can be rendered at different sizes: tiny objects can be made larger for more detailed observation, and huge objects can be made smaller for a better big picture. The same structure can have different appearances at different scales, informing users with the rich features of the structure, as seen in Figure 5. The understanding of a complicated structure could be improved.

3.1.2 Cross-Scale Information Sharing. Collaboration can facilitate people's work by improving communications and/or cooperation on shared artifacts (Dix, 1994). Multiscale collaboration focuses more on artifacts than communications. In such artifact-centered collaboration, users often need to refer to shared artifacts (Dix). Conventional CVEs are usually "What-You-See-Is-What-I-See" (WYSIWIS) or "relaxed WYSIWIS" (Stefik, Bobrow, Foster, Lanning, & Tatar, 1987), so knowing what others are referring to is fairly easy. mCVEs are not like this, however. Users may have very different views at different scales, and see totally different objects and structures. Figure 7 shows the view of a city planner who oversees the whole city (a), and the view of another planner who checks buildings at the street level (b). These two views, which are from very



(a) (b)

Figure 7. *Two views of a city at two scale levels.*

different perspectives and show different levels of abstraction of the city, may make it difficult for two planners to understand how their work can be related. A planner can refer to a particular object in her view, but the other may not see it at all. Multiscale technology may hurt mutual understanding of objects in collaboration because of scale-related views.

3.1.3 Cross-Scale Action. Multiscale can also bring some benefits to collaboration. One of them is that users can leverage their different action capabilities and different scopes of influence of their actions. Take the example of the two city planners again. When the street-level planner needs to go to a distant place, the city-level planner, with a great action domain, can move the partner easily to the destination and reduce the task time. Or when the city-level planner wants to move to an exact location, the street-level planner can move the partner's avatar accurately because of high manipulation accuracy. It is mentioned previously that a similar technique can be used in object manipulation. Thus, multiscale collaboration can improve navigation and manipulation activities by allowing users to take advantage of each other's different action capabilities.

3.2 Design Considerations

Since multiscale technology could affect user interaction both positively and negatively, the design of mCVEs focuses on maximizing the advantages of multiscale technology and avoiding the disadvantages. This principle leads to the following design considerations.

3.2.1 Support for Better Understanding of Multiscale Structures. Multiscale structures should be modeled in such a way that scale-dependent characteristics can be observed at different scale levels. In the real world, multiscale structures are ubiquitous, as seen in Figures 1 and 2. To observe these multiscale structures, we need only appropriate tools. In 3D virtual worlds, however, what users can see is predesigned. Thus, to visualize multiscale structures, objects have to be modeled and rendered in a multiscale way.

3.2.2 Support for Cross-Scale Information Sharing. Users should be able to share artifacts of common interest with others across scale. Tools such as multiple views of the world (Gaver, Sellen, Heath, & Luff, 1993) cannot help too much. Such techniques require some common objects in different views as references, but users' very subjective views in mCVEs may make it difficult to find such references. It is a challenge to build mutual understanding of working contexts with subjective views (Snowdon & Jää-Aro, 1997). Users need help in understanding how their subjective views and objects in them are related to each other.

3.2.3 Support for Cross-Scale Action. Users should be allowed to get involved in others' work across scale so that they can leverage their different interaction capabilities in collaboration. They should be able to directly manipulate objects that others are working on, and move the avatars of other users. Of course, this intervention should be based on mutual agreements to avoid any unwanted consequences. In the city-planning example, the city-level planner should not change things the street-level planner is working on without consent. Otherwise, the street-level planner may get confused and not understand what has happened.

3.3 Conceptual Design

Based on these considerations, some multiscale tools were designed. Here, discussions focus on designs that are fundamentally important to mCVEs, including scale-based semantic representations that present scale-related characteristics of multiscale structures, dynamic

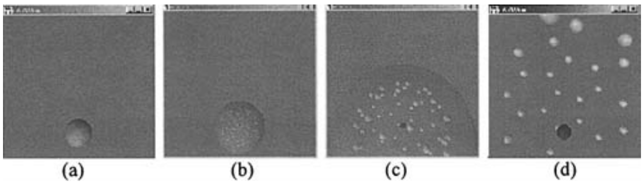


Figure 8. *Successive fading views in semantic representations.*

view transition that facilitates cross-scale information sharing, and direct avatar manipulation that supports cross-scale action.

3.3.1 Scale-Based Semantic Representations for Multiscale Structures. This design renders a multiscale structure with different representations at different scales. Figure 5 presents the visual results of a substance under this technique. Initially, the substance appears as a double helix in (a). The user changes the scale to increase the structure size and to see more details in (b). Continuing the scaling, the user finds a different structure in an atom in (c). The user is informed of semantic characteristics of the substance at different scales.

This scale-based representation technique has one problem. The change between two very different representations may confuse users. For example, the sudden appearance of the structure in Figure 5(c) may distract users from their primary tasks. To address this issue, we adopted a fading tool (Bederson & Hollan, 1994; Lieberman, 1994) to smooth the transition between different representations. Figure 8 shows a transition process from a spherelike molecule from Figure 8(a) to a set of atoms inside it in Figure 8(d). Instead of jumping abruptly between different representations, the disappearance of the molecule and the appearance of atoms are gradual. The transition between two representations is more easily comprehensible. Cognition required in understanding the change of visual image can be kept at the level of perceptual processing so that users can free their cognitive processing capacities for more complicated tasks (Robertson, Card, & Mackinlay, 1991).

Scale-based semantic representations can be seen as one member of the family of level of detail (LOD) techniques, which visualize the same structure with different geometric objects according to a particular interaction parameter, but with a very different purpose. While most LOD techniques such as distance LOD primarily concern *computation efficiency* (Puppo & Scopigno, 1997), scale-based semantic representation gives more consideration to the user's *interaction needs*. Common LOD techniques render individual objects with different textures or geometries, but scale-based semantic representation visualizes structures with different semantic abstractions. LOD techniques can be integrated into scale-based semantic representations to improve both user interaction and machine performance. For example, in a city-planning project, a city can have different structures at the level of city, district, and street to inform planners semantically, and distance LOD can be applied in rendering individual objects in each structure to improve 3D computation.

3.3.2 Dynamic View to Support Cross-Scale Information Sharing. Design here addresses issues concerning how to help users at different scales to build common understandings of their divergent views. One design choice is to use a dynamic view to narrow down the view difference. We used animation to show the transition between two views. Animation is generated by interpolating the view positions, view orientations, and view scales of two views. For example, if two users' views at two different scales look like Figures 7(a) and 7(b), an animation can be generated by adding intermediate image frames between Figure 7(a) and 7(b). The animation can tell how one view can be transformed to another, and how objects in two views are related to each other.

3.3.3 Direct Avatar Manipulation to Support Cross-Scale Action. To help users get involved in others' work so that they can take advantage of their different action capabilities, we adopted a design choice that allows direct manipulation on other users' avatars. With such a tool, planners in the above city-planning example can easily cooperate on tasks that require large action domain and high action accuracy. Of course, this direct interference should be regulated. The user who owns the avatar is always aware of such action and has control over whether this action is allowed.

The implication of such a direct manipulation of avatar body for user interface design deserves more attention. Under this design, an avatar can "afford" manipulation activities. This is quite different from the traditional role of an avatar, merely as user embodiment, in conventional CVEs. The reason that no direct manipulation of an avatar is allowed in conventional CVEs might be due to the influence of real life, where the autonomy of human bodies is well respected. To better facilitate people's work in virtual environments, some real-life constraints could be relaxed or even abandoned. Virtual environments should not be treated as the replication of the real world, just as new technologies should not be interpreted by old metaphors. Direct manipulation of a user's avatar by others is an effort to break down the metaphor of reality.

4 System Implementation

A desktop mCVE prototype system consisting of over 20,000 lines of Java code was implemented. Discussions in this section focus on the choice of toolkits in software implementation, general system architecture, and the implementation of some multiscale tools.

4.1 Choice of Toolkits

The design of virtual environments that support collaboration needs to consider issues in three aspects: 3D graphics rendering, network communication management, and user interactions (Singhal & Zyda, 1999). It is not our interest to address issues related to low-level graphics rendering or network communications. Our goal here is to support user interactions with multiscale structures and with other users at different scales. To achieve this interaction-oriented design goal, we implemented the mCVE prototype with two toolkits: Java 3D and Java Shared Data Toolkit (JSDT).

This hybrid approach allows a design focus on high-level interaction tools, because Java 3D can handle graphics rendering and JSDT can handle network communications issues. With the scene-graph management of Java 3D, we can construct virtual environments and structures at the level of geometric objects, rather than at the level of vertices and polygons. Java 3D allows the extension of new rendering behaviors, which is not well supported by other CVE toolkits such as DIVE (Carlsson & Hagsand, 1993) and SPLINE (Waters et al., 1997). This flexibility is critical to the implementation of such tools as scale-based semantic representations and dynamic view transition. Furthermore, the easy integration of other Java APIs, such as Java Swing, into Java 3D can simplify user interface design. JSDT is ideal for the research because it provides flexible data delivery services in support of synchronous collaboration on the Internet. JSDT supports basic abstraction of network sessions, multicast message communications, and concurrency control. Thus, the design can focus on optimizing the management of distributed 3D scenes and cross-scale communications. One drawback of this hybrid approach is the lack of those CVE services seen in SPLINE (Waters et al.). Collaboration services have to be built from scratch.

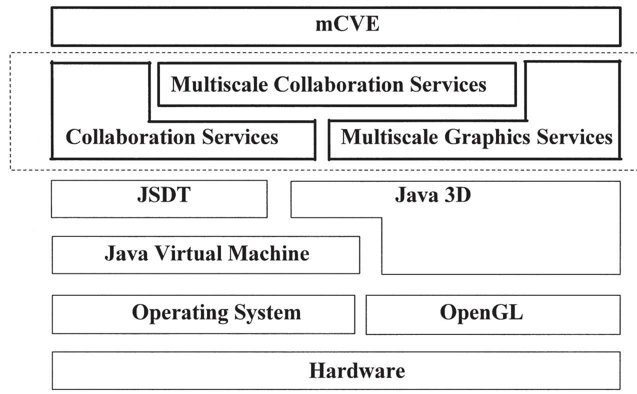


Figure 9. *System layer of the mCVE prototype.*

4.2 System Layer and Architecture

The system layer of the prototype is shown in Figure 9. The implementation focused on the two layers inside the dashed rectangle. On top of Java 3D and JSDT are multiscale graphics services and collaboration services, and above them are multiscale collaboration tools.

Multiscale graphics services handle object-rendering issues and update user views based on their view positions and interaction scales. Collaboration services deliver messages among distributed users and manage the consistency of shared virtual environments. Multiscale collaboration services deal with such collaboration issues as cross-scale information sharing and cross-scale avatar manipulation. These services lay the foundation for multiscale tools.

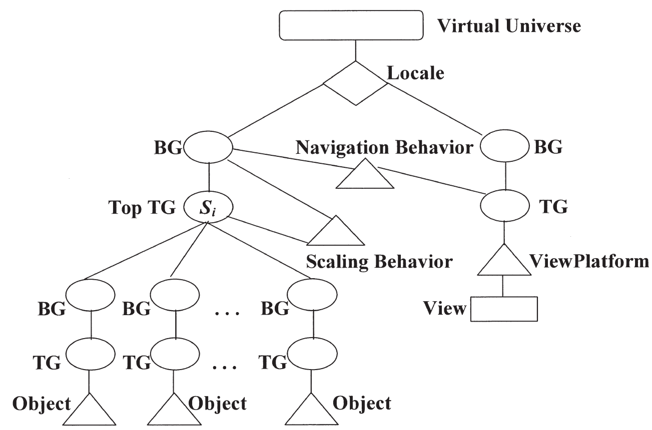


Figure 10. *Scene graph of the prototype system.*

4.3 Implementation of Multiscale Tools

A set of multiscale collaboration tools were implemented and integrated into the prototype system. This section introduces such tools as scaling control, which distinguishes mCVEs from other CVEs, scale-based semantic representations, which visualize information in a multiscale way, cross-scale dynamic view transition, which addresses the cross-scale information problem uniquely seen in multiscale collaboration, and the management of avatar position in direct avatar manipulation, which illustrates cross-scale data coordination.

4.3.1 Scene Graph of the Prototype System and Scaling Control. The scene graph of the system is shown in Figure 10. Each Java 3D scene graph has a Virtual Universe object as the root. A Locale object attaches scene graphs to the Virtual Universe object. BranchGroup (BG) nodes collect such objects as geometry, behavior, light, sound, and so forth. A TransformGroup (TG) node defines the coordinates of objects under it.

The system has two subscene graphs: the object scene graph, which is the left child of the Locale object, and the view scene graph, which is the right child. The object scene graph collects all objects and behavior nodes. The view scene graph defines the viewing behaviors.

To change a user's interaction scale is to change the values of the user's interaction parameters relative to the size of the virtual world. Thus, scale change can be implemented in two ways: (1) scaling the virtual world up and down and keeping all interaction parameters untouched; or (2) changing all interaction parameters and keeping the virtual world untouched. Mathematically, they are equivalent. We adopted the first approach, because scaling the virtual world up and down can be done by modifying just one parameter, the scaling factor, S_i, of the top TG in the object scene graph. This is much simpler than the second approach, which involves the modification of complicated view parameters.

4.3.2 Scale-Based Semantic Representations. In the implementation of scale-based semantic representations, we adopted a design that uses a delegate object to wrap all representations and that delivers appropriate representations based on the value of the controlling variable (Fox, 1998). A scale-based semantic representation node is extended from the Java 3D Switch object. Each node has a set of child representations and an ordered array that defines the scale range for each representation (Figure 11).

4.3.3 Animating View Transition in Collaborative View Sharing. The key issue in making a view-transition animation is to interpolate two views. In mCVEs, a view can be uniquely determined by its view position $\vec{P}$, its view orientation $\vec{O}$, and its view scale S, and be written as $V(\vec{P}, \vec{O}, S)$.

Given two views $V_0 = V(\vec{P}_0, \vec{O}_0, S_0)$ and $V_1 = V(\vec{P}_1, \vec{O}_1, S_1)$, to implement a view-transition animation is to

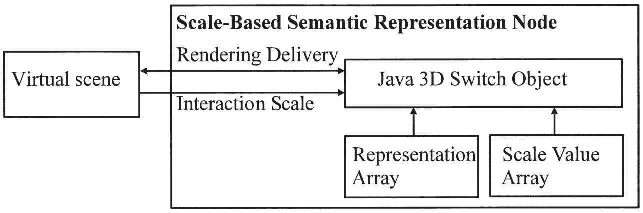


Figure 11. *Scale-based semantic representation node.*

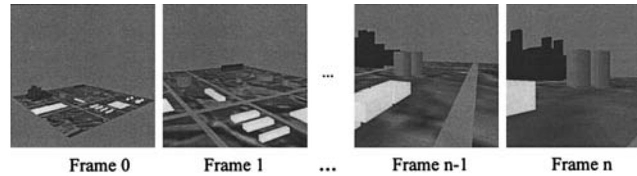


Figure 12. *Frames of a dynamic view-transition animation.*

choose a path that links V_0 and V_1 and to set the trajectory of viewpoint movement, view direction along the path, and observation scale along the path. View-path and view-orientation interpolation techniques are mature (Parent, 2001; Watt & Policarpo, 2001), so this research focused only on the interpolation of view scale.

A scale-interpolation function could be linear, logarithmic, or in other forms. We chose a logarithmic function, because it gives users a constant relative rate of change in their views. Interpolating the view scale logarithmically can give users a familiar experience.

Assume the animation includes n frames from V_0 to V_1. The view scale of the ith frame ($0 \leq i \leq n$) in a logarithmic interpolation function can be written as:

$$S_i = (\Delta S)^i \cdot S_0 \quad (1)$$

where:

$$\Delta S = \left(\frac{S_1}{S_0}\right)^{1/n} \quad (2)$$

With corresponding $\vec{P}_i$ and $\vec{O}_i$ from interpolation functions for view position and view orientation, the view of the ith frame, $V_i = V(\vec{P}_i, \vec{O}_i, S_i)$, can be obtained. Figure 12 shows two views seen in Figure 7, as Frame 0 and Frame n, and other interpolated view frames. (Due to space limitations, we have shown only four frames used for the animation.) Showing these frames successively with a reasonable frame rate can produce an animation linking these two divergent views.

4.3.4 Cross-Scale Data Consistency and Avatar Position in Direct Avatar Manipulation.

Although changing the scaling factor of the top TG in the object scene graph simplifies the scaling operation, such an approach poses a challenge in cross-scale data consistency. In CVEs, all users usually need to see the same object appearing at the same spatial location. In conventional CVEs, this can be easily achieved, because an object's coordinates in different users' worlds are often the same. In mCVEs, however, when users are at different scales, their virtual worlds are rescaled according to the scaling factors of the top TG in their own object scene graphs. The same object will have different local coordinates in different users' worlds. When the object is moved by a user, the object's local coordinates in this user's world cannot be used to update the object's new position in all other users' worlds.

In the implementation, we used normalized coordinates, which are calculated as the ratio of the object local coordinates in the user's world to the user's scale. Let $P_l(x_l, y_l, z_l)$ and S_l denote the local coordinates and the user's scale respectively. The normalized coordinates, P_n, can be written as:

$$P_n = P_l / S_l \quad (3)$$

By doing so, the same object will always have the same P_n. This P_n is used to define the coordinates of the object in every user's object scene graph. Modifying the spatial location of an object in a user's world leads to a new P_n. All users can use this P_n to update the coordinates of the object in their object scene graphs. Therefore, P_n enforces data consistency across scale. Because the scaling factor is on the top of the scene graph, the object's final location in a user's world is P_l, the product of P_n and this user's scale S_l. With different S_l, users will be able to see the world rendered at different scales.

An avatar's position can be managed similarly, but with a small difference. An avatar object differs from other ordinary objects in a way that in its host user's

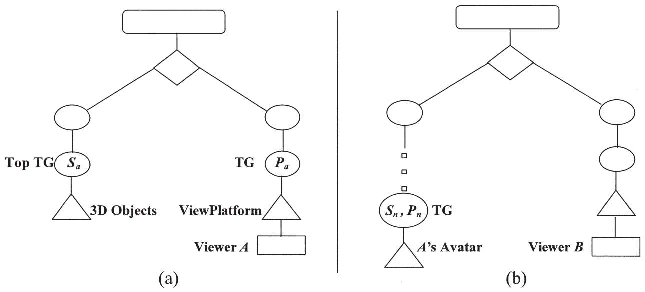


Figure 13. *Two scene graphs for the same avatar.*

world, the coordinates of the avatar are affiliated with the view scene graph, rather than the object scene graph, so that the avatar always moves with the view. In other users' worlds, however, the avatar object is under the object scene graphs. Figure 13(a) is the scene graph of the world of a user, say *A*, who interacts with the virtual environment at a location, P_a, and with a scale, S_a; Figure 13(b) is the scene graph of the world of another user, say *B*. User *A*'s avatar is in User *B*'s object graph. Because P_a is not governed by S_a, the normalized coordinates of the avatar in User *B*'s object graph should be written as:

$$P_n = P_a / S_a \tag{4}$$

The body size of User *A*'s avatar changes with the scale of User *A*, so a scale parameter is also needed to define the size of User *A*'s avatar in User *B*'s object scene graph so that User *B* can be aware of User *A*'s scale. This scale should also be normalized. Let S_n denote the normalized scale, where S_n is simply the reciprocal of S_a:

$$S_n = 1/S_a \tag{5}$$

This is because the avatar body changes inversely proportional to the scale factor S_a (shrinking the world is equivalent to growing the avatar body, and growing the world is equivalent to shrinking the avatar body).

The implementation of other cross-scale tools, including direct avatar manipulation, also uses P_n and S_n to define the coordinates and size of an avatar entity. This approach guarantees the consistency of the avatar position and size in all users' views at different scales.

Table 1. *Six Treatments in a 2 × 2 + 2 Design*

	Noncollaboration	Collaboration
Nonmultiscale	VE	CVE
Multiscale	M-VE	NR
		GUIDE
		MOVE

5 Experimental Evaluation

We evaluated multiscale collaboration in subject tests. An experiment was designed to assess the benefits users can get from an mCVE in accomplishing tasks requiring cross-scale coordination.

5.1 Experiment Design and Procedure

Subject tasks in the experiment involved searching for a "bomb" on a square ground plane (2000 × 2000 m^2), with a distinctly shaped building in each corner (square, hexagon, octagon, and circle as seen from above). Each had a height of 12 m and a base of about 6400 m^2. On the ground behind each building was a unit cube (1 m^3) containing a unique text name and smiley face (about 0.5 × 0.5 m^2). The bomb was inside one of these four cubes. In the test, subjects were placed in the middle of the square plane. They had a default eye level of 1.68 m and a default moving step of 1 m in the virtual environment. The shape of the building that the bomb was nearby was known in advance, and subjects needed to find that building, locate the bomb box, and key in the name of the box to defuse the bomb.

A 2 × 2 + 2 factorial design that has six treatments, two noncollaboration and four collaboration, was adopted (Table 1). For the two noncollaboration treatments, VE is just a conventional 3D virtual environment, and M-VE is a VE enhanced by multiscale tools; specifically it allows users to change their interaction scales, eye level, and speed. Among four collaboration treatments, CVE, a conventional collaborative virtual environment, is the only one without multiscale tools. The other three treatments, all equipped with multiscale tools, differ in the assignment of subject task roles (being a giant or a normal person) and in the way subjects affect each other's work across scales. In one treatment, the roles of subjects are not predefined, and subjects can choose their own interaction scales as desired. This environment is labeled as NR (No Role). The other two mCVEs both assign one subject to be a giant and the other a normal person so that subjects can change their interaction scales only within a limited range. In one such environment, the giant is permitted to move the partner directly, and this treatment is labeled as MOVE. Another condition allows the giant to guide the movement of the partner only verbally, and is denoted as GUIDE.

Recruited through email, 24 students paired in 12 groups participated in the experiment. In noncollaboration treatments, they worked on their own, and in collaboration treatments, they communicated through an audio channel. The performances were measured by task-completion time.

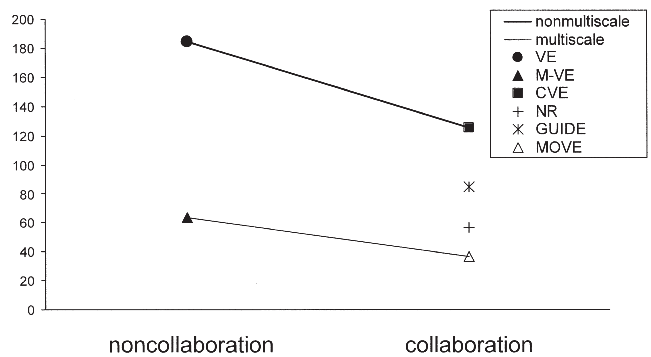


Figure 14. *Time (in seconds) for the six treatments.*

5.2 Results

An ANOVA analysis of data from four treatments (VE, M-VE, CVE, and MOVE) shows main effects of both collaboration ($F_{1,70} = 12.98$, $p < 0.001$) and multiscale ($F_{1,70} = 70.90$, $p < 0.0001$). Interaction is not significant ($F_{1,70} = 1.87$, $p = .176$), as seen in Figure 14. Subjects performed best in MOVE, where they could take full advantage of multiscale and collaboration, and they did worst in VE, where there was no help at all.

Subjects used different strategies in the different conditions. Without multiscale tools, subjects had to go around and count the number of sides of all four buildings to find the target, a very time-consuming process. With multiscale, subjects could increase their sizes to see building shapes from above and approach the bomb quickly, so the time required was be reduced significantly.

The performance difference among the three mCVEs also indicates the importance of different collaboration supports. In the NR treatment, subjects had to assume their roles via an expensive negotiation process. In the GUIDE treatment, the giant subject, who could see the building shape, must coordinate with the partner about navigation verbally through a costly grounding process (Clark & Brennan, 1991). In MOVE, however, the all-seeing giant could actually move the partner to the destination quickly, and bypassed grounding processes.

6 Discussion

In this paper, we have introduced a novel interactive environment, multiscale collaborative virtual environment, which gives users the freedom to control their interaction scales in collaboration. Multiscale collaboration can help better allocate scarce scale resources and combine different expertise and action capabilities of individual users in their work with large structures. We have also discussed the implications of multiscale technology for user interaction and the design and development of a Java-based mCVE system. Our subject tests have shown that multiscale collaboration can facilitate people's work that requires cross-scale information and collaboration.

Research reported in this paper has largely focused on general interaction issues in mCVEs. Designs have been limited to generic tools to support multiscale collaboration, and experimental study has focused on general tasks. We expect that multiscale collaboration tools can be used to support people's specific work in the real world. To achieve this goal, the design of multiscale tools should support domain-specific tasks. One of our ongoing research projects focuses on studying how multiscale collaboration can support research in materials science and engineering. We are investigating the needs of materials scientists for multiscale collaboration, and will design and implement tools that support these needs. The effectiveness and usability of these tools will also be evaluated when they are ready.

At the same time, we are interested in behavior issues in mCVEs. Some important research questions concerning user behavior in mCVEs have not been studied thoroughly. It is still unclear when and how users can benefit from collaborative object manipulation across scale, whether the designed dynamic view-transition technique can indeed improve cross-scale information sharing, what mechanism is needed to help users better manage cross-scale collaboration overheads, and so forth. In addition, it has been found that multiscale also brings some new challenges in social interactions (Zhang & Furnas, 2002), and efforts are needed to study what can be done to address these interaction issues. Research in this direction will deepen our understanding of mCVEs, and help to design better tools to support multiscale collaboration.

Acknowledgments

This research was funded in part by Microsoft Research.

References

Ahl, V., & Allen, T. (1996). *Hierarchy theory: A vision, vocabulary and epistemology.* New York: Columbia University Press.

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language.* Oxford, UK: Oxford University Press.

Bederson, B., & Hollan, J. (1994). Pad++: A zooming graphical interface for exploring alternate interface physics. *Proceedings of ACM UIST '94,* 17–26.

Benford, S., Bowers, J., Fahlén, L., Greenhalgh, C., & Snowdon, D. (1995). User embodiment in collaborative virtual environments. *Proceedings of ACM CHI '95,* 242–249.

Benford, S., Snowdon, D., Colebourne, A., O'Brien, J., & Rodden, T. (1997). Informing the design of collaborative virtual environments. *Proceedings of ACM GROUP '97,* 71–80.

Carlsson, C., & Hagsand, O. (1993). DIVE—A platform for multi-user virtual environments. *Computer and Graphics, 17*(6), 663–669.

Clark, H., & Brennan, S. (1991). Grounding in communication. In L. B. Resnick, J. Levine, & S. D. Behrend (Eds.), *Perspectives on socially shared cognition* (pp. 127–149). Washington, DC: American Psychological Association.

Dix, A. J. (1994). Computer-supported cooperative work—A framework. In D. Rosenburg & C. Hutchison (Eds.), *Design Issues in CSCW.* (pp. 23–37). Berlin: Springer-Verlag.

EPFL, Geneva, IIS, TNO, Lancaster, UCL, et al. (1997). *Guidelines for building CVE applications.* COVEN Project. Retrieved November 21, 2002, from http://www.cs.ucl.ac.uk/staff/m.slater/vr/Coven/PublicDeliverables/del26.PDF.

Fox, D. (1998). *Tabula rasa: A multi-scale user interface system.* Unpublished doctoral dissertation, New York University.

Furnas, G. (1986). Generalized fisheye views. *Proceedings of ACM CHI '86,* 16–23.

Gaver, W., Sellen, A., Heath, C., & Luff, P. (1993). One is not enough: Multiple views in a media space. *Proceedings of ACM INTERCHI '93,* 335–341.

Hollan, J., & Stornetta, S. (1992). Beyond being there. *Proceedings of ACM CHI '92,* 119–125.

Koike, H., & Yoshihara, H. (1993). Fractal approaches for visualizing huge hierarchies. *Proceedings of IEEE VL '93,* 55–60.

Lamping, J., Rao, R., & Pirolli, P. (1995). A focus + context technique based on hyperbolic geometry for visualizing large hierarchies. *Proceedings of ACM CHI '95,* 401–408.

Leigh, J., Johnson, A., & DeFanti, T. (1996). CALVIN: An immersimedia design environment utilizing heterogeneous perspectives. *Proceedings of IEEE International Conference on Multimedia Computing and Systems,* 20–23.

Lieberman, H. (1994). Powers of ten thousand: Navigating in large information spaces. *Proceedings of ACM UIST '94,* 15–16.

Mackinlay, J., Card, S., & Robertson, G. (1990). Rapid controlled movement through a virtual 3D workspace. *Proceedings of ACM SIGGRAPH '90,* 171–176.

Mine, M., Brooks, F., & Sequin, C. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. *Proceedings of ACM SIGGRAPH '97,* 19–26.

Morrison, P., & Morrison P. (1994). *Powers of ten: About the relative size of things in the universe, and the effect of adding another zero.* San Francisco: W. H. Freeman.

Parent, R. (2001). *Computer animation: Algorithms and techniques.* San Francisco: Morgan-Kaufmann.

Pattee, H. (1973). *Hierarchy theory: The challenge of complex systems.* New York: George Braziller.

Perlin, K., & Fox D. (1993). Pad: An alternative approach to the computer interface. *Proceedings of ACM SIGGRAPH '93,* 57–64.

Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., & Mine, M. (1997). Image plane interaction techniques in 3D immersive environments. *Proceedings of the 1997 Symposium on Interactive 3D Graphics,* 39–43.

Poupyrev, I., Billinghurst, M., Weghorst, S., & Ichikawa, T. (1996). The go-go interaction technique: Non-linear mapping for direct manipulation in VR. *Proceedings of ACM UIST '96,* 79–80.

Puppo, E., & Scopigno, R. (1997). Simplification, LOD, multiresolution: Principles and applications. *Proceedings of EUROGRAPHICS, 16* (3).

Raab, A., & Ruger, M. (1996). 3D-Zoom: Interactive visualization of structures and relations in complex graphics. In B. Girod, H. Nieman, & H. Seidel (Eds.), *3D image analysis and synthesis* (pp. 125–132). Sankt Augustin, Germany: Infix-Verlag.

Robbins, M. (2001). Modeling and simulation of the failure and toughness of polymer/substrate bonds. *The second Annual Multiscale Modeling—Bridging the Gap Between Atomistic and Meso Scales.*

Robertson, G., Card, S., & Mackinlay, J. (1991). Information visualization using 3D interactive animation. *Proceedings of ACM CHI '91,* 461–462.

Robertson, G., & Mackinlay, J. (1993). The document lens. *Proceedings of ACM UIST '93,* 101–110.

Robertson, G., Mackinlay, J., & Card, S. (1991). Cone trees: Animated 3D visualizations of hierarchical information. *Communications of the ACM, 34* (2), p. 89–194.

Sarkar, M., & Brown, M. H. (1992). Graphical fisheye views of graphs. *Proceedings of ACM CHI '92,* 83–91.

Singhal, S., & Zyda, M. (1999). *Networked virtual environments: Design and implementation.* New York: ACM Press.

Snowdon, D., & Jää-Aro, K. (1997). A subjective virtual environment for collaborative information visualization. *Virtual Reality Universe '97.* Retrieved May 23, 2003, from http://www.cyberedge.com/vru_papers/snowden.htm.

Stappers, P., Gaver, W., & Overbeeke, C. (2000). Beyond the limits of real-time realism, 2000. Chapter to appear in L. J. Hettinger & M. W. Haas (Eds). *Psychological issues in the design and use of virtual and adaptive environment.* Mahwah, NJ. Erlbaum.

Stefik, M., Bobrow, D., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revised: Early experiences with multiuser interfaces. *ACM Transactions on Information Systems (TOIS), 5*(2), 147–167.

Stoakley, R., Conway, M., & Pausch, R. (1995). Virtual reality on a WIM: Interactive worlds in miniature. *Proceedings of ACM CHI '95,* 265–272.

Sutherland, I. (1965). The ultimate display. *Proceedings of IFIP Congress 2.*

Tromp, J. G. (1999). *Usability design for CVEs.* COVEN Project. Retrieved November 21, 2002, from http://coven.lancs.ac.uk/4/deliverables/del29.pdf.

Waters, R., Anderson, D., Barrus, J., Brogan, D., Casey, M., McKeown, S., et al. (1997). Diamond park and SPLINE: Social virtual reality with 3D animation, spoken interaction, and runtime extendibility. *Presence: Teleoperators and Virtual Environments, 6*(4), 461–481.

Watt, A., & Policarpo, F. (2001). *3D games: Animation and advanced real-time rendering.* New York: Addison-Wesley.

Zhang, X., & Furnas, G. (2002). Social interactions in multiscale CVEs. *Proceedings of ACM CVE 2002,* 31–38.